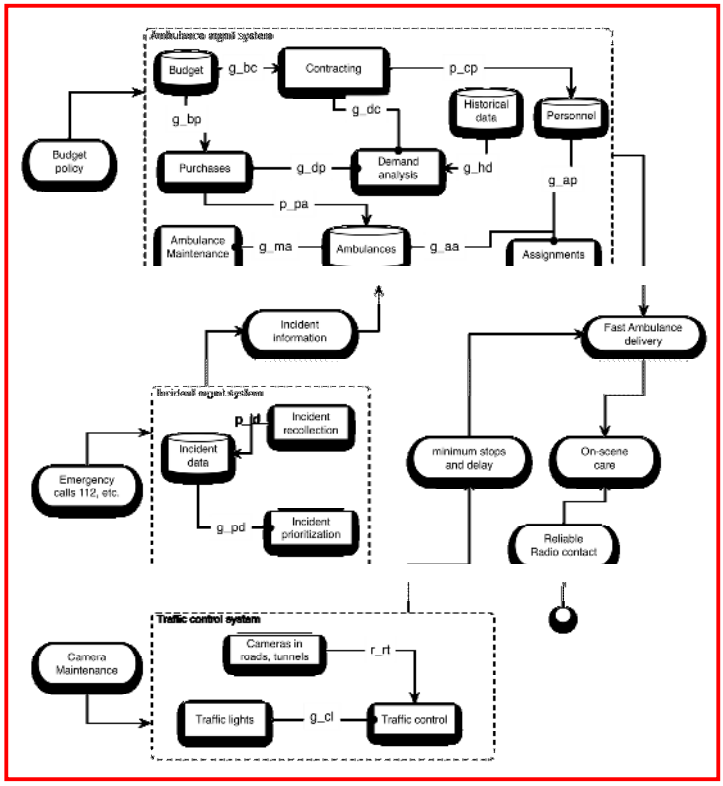
JRC Scientific and Technical Reports

Infrastructure (Resilience-oriented) Modelling Language: I®ML

A proposal for modelling infrastructures and their interconnections

Andrés Silva, Roberto Filippini



EUR 24727 EN - 2011

JRC EUROPEAN COMMISSION

ipsc Institute for the Protection and Security of the Citizen

The mission of the JRC-IPSC is to provide research results and to support EU policy-makers in their effort towards global security and towards protection of European citizens from accidents, deliberate attacks, fraud and illegal actions against EU policies.

European Commission
Joint Research Centre
Institute for the Protection and Security of the Citizen

Contact information
Address: TP 210, EC JRC Ispra, Ispra (Va) Italy
E-mail: roberto.filippini@jrc.ec.europa.eu
Tel.: +39 0332 789936
Fax:

http://ipsc.jrc.ec.europa.eu/
http://www.jrc.ec.europa.eu/

JRC 63302

EUR 24727 EN
ISBN 978-92-79-19324-8
ISSN 1018-5593
doi:10.2788/54708

Luxembourg: Publications Office of the European Union

Printed in Italy

Infrastructure (Resilience-oriented) Modelling Language: I®ML
A proposal for modelling infrastructures and their connections

Andrés Silva[1]
Universidad Politécnica de Madrid

Roberto Filippini
JRC of the European Commission

Abstract

The modelling of critical infrastructures (CIs) is an important issue that needs to be properly addressed, for several reasons. It is a basic support for making decisions about operation and risk reduction. It might help in understanding high-level states at the system-of-systems layer, which are not ready evident to the organisations that manage the lower level technical systems. Moreover, it is also indispensable for setting a common reference between operator and authorities, for agreeing on the incident scenarios that might affect those infrastructures. So far, critical infrastructures have been modelled ad-hoc, on the basis of knowledge and practice derived from less complex systems. As there is no theoretical framework, most of these efforts proceed without clear guides and goals and using informally defined schemas based mostly on boxes and arrows. Different CIs (electricity grid, telecommunications networks, emergency support, etc) have been modelled using particular schemas that were not directly translatable from one CI to another. If there is a desire to build a science of CIs it is because there are some observable commonalities that different CIs share. Up until now, however, those commonalities were not adequately compiled or categorized, so building models of CIs that are rooted on such commonalities was not possible. This report explores the issue of which elements underlie every CI and how those elements can be used to develop a modelling language that will enable CI modelling and, subsequently, analysis of CI interactions, with a special focus on resilience.

Keywords: critical infrastructures, system of systems, interdependencies, emergent behaviour, control systems, safety, risk, resilience.

[1] Universidad Politécnica de Madrid, Campus de Montegancedo, s/n. 28660. Boadilla del Monte, Madrid, Spain, Telephone: +34 91 336 6921. Fax: +34 91 336 6917. E-mail: asilva@fi.upm.es Ground work for this document was elaborated under the contract order N. CCR.IPSC.B255717

Executive Summary

Civil and industrial installations do not work in isolation, and in many cases they form complex networked systems of systems (SoS). Examples are modern infrastructures for the distribution of energy (power grids), ICT networks and transportations, civil emergency services. A network of SoS can be conceptually conceived as an open architecture. The elements interconnect as long as they possess the right requisites of interoperability. Nonetheless, several problems of integration exist when dealing with systems that are heterogeneous and even more subtle issues come into play when considering hazards and accident scenarios.

The design and assessment of the networked systems is difficult and often beyond the capabilities of traditional engineering. SoS and infrastructures present unique features, and existing tools result inadequate, especially for cross-sector analysis. The subject has stimulated research interest in diverse directions, among which the emerging topic is by far the modelling and analysis of network interdependencies.

In this report, we present a language for modelling networked infrastructures. The goal of the language is to identify the consequences of vulnerabilities in combination with the ability of the network to face/resist different threats, i.e. its resilience. Most accident scenarios in networked systems cannot be addressed by a simplistic black or white (i.e. functioning or failed) approach. Slow deviations from nominal operation conditions may cause degraded behaviours that suddenly end up into unexpected malfunctioning, with large portions of the network affected. The language makes it possible the identification of these accident scenarios, by representing the propagation of failure events throughout the network. A key feature of the language stands in the capacity to represent interdependencies of various natures, e.g. technical, organizational and human. This representation also includes control loops with physical quantities and related information. The resulting model can be used for assessing the effectiveness of the protections measures that contribute to the overall resilience, both in qualitative and quantitative terms.

A few case studies are provided in order to show how the language can be applied and the expected advantages. The presented methodology lends itself to be applied in combination with already existing system analysis techniques, such as risk assessment, dependability and performance evaluation.

Table of Contents

1 Introduction

Modern infrastructures in transportation, energy and communications are experiencing a continuous growth (in size, interconnections, and integration) that responds to the demand of having ubiquitous services. This process takes benefits to the society and economy but it is not immune to risks. The existence of these risks justifies infrastructures to be considered critical under many viewpoints.

A Critical Infrastructure (CI) is a complex assembly of systems, i.e. a system-of-systems, where equipments, people and resources, in different independent organisations, interact, cooperate, exchange and dispatch services at various levels. The distinctive features of a critical infrastructure emerge when looking at the way it is designed and operated. An infrastructure is not the result of a preconceived "design", rather it is the product of a bottom-up process that ends up in the aggregation of its parts without an actual socio-technical reorganization. Management and decisions are in this context under the responsibility of a multiplicity of actors (operators, utilities, governments), which are sensitive to diverse types of priorities, technical, economic or societal. Information and Communication Technologies (ICT) act –in most milieus– as the backbone, facilitating controls and operations at a large scale (i.e. the electric power grid). This role of ICT represents a risk per-se. For instance, it leads to a resulting tight coupling of infrastructures that is the vehicle for the fast propagation of degraded behaviour throughout the network, across the organisational borders. This latter aspect concerns the protection of assets in face of risks that, in absence of a global harmonized view of the problem, gives origin to conflicts among corporate, national and cross-national interests. Given the premises, it is evident that the study of a critical infrastructure is a very complex subject. The challenge is about finding instruments to master the complexity [Sou08, Geo10], and at the same time to represent the information required for decision makers at the needed level of detail [Mas10, Val08].

The rational organization of the subject, from a theoretical point of view, has its fundaments on the general theories of reductionism and holism. The "reductionism" standpoint says that an infrastructure is an assembly of parts, which can be analysed separately and recomposed in the global picture. The "holism" position states that the infrastructure is a non-decomposable entity, whose behaviour cannot always be inferred from the separated representation of its parts. The dualism holism versus reductionism offers interesting and stimulating ideas for research [Hey89, Joh06, Hai08]. Holistic theories focus on the behavioural aspect of the infrastructure, and postulate the existence of *emerging behaviours*, as the result of interconnections, interdependencies and also control feedbacks among the various parts, resulting in non-linear, unpredictable dynamics. The supporters of reductionism take a conservative position in this respect, assuming that in most respects the understanding of the whole system-of-systems can be obtained by evaluating the components and their interactions. Anyway, it is interesting to look at the objections that they bring. The most important is that engineering is able to realize artefacts that must behave as expected, and this holds for simple to more complex systems, such as networked infrastructures. They enforce this statement by saying that there is a phase where technology always struggles to catch up the demand of the society, and during this period the products may result unstable and unsatisfactory. Another element to be considered is the inadequacy of tools at disposal to represent the problem [Lev04, Holl06].

Despite the theoretical positions, a methodological gap still exists when dealing with the representation (i.e. modelling) of critical networked infrastructures. The modelling of CIs goes into two directions: 1) structure-architecture-topology, and 2) systemic-behavioural. The structural properties of an infrastructure have received more attention so far – not the least due to the relative easiness to capture them. This reflects the need of representing the topology of the network with its interconnections and links, but also the threats and vulnerabilities affecting its components [Rin01, Pan08, Bom09, Cas08,

Lap07, Ouy09, Joh09]. The behavioural representation of an infrastructure is able to return more insights, especially when the goal is the representation of systemic attributes, such as safety (and risk) and end-to-end performance. Indeed, while models for reliability and security can be mostly derived from the structural representation of systems, systemic attributes cannot be inferred in this way, but they need further insights on the behaviour.

It is also important to remark that critical infrastructures suffer from a vast set of hazards that can be only partially identified with the traditional instruments of accident analysis. Besides internal component failures, there are also drifts to changes, variability in the operation conditions and dysfunctional failures. These hazards are either not caused by faults (e.g. dysfunctional failures), or observable at the system-of-systems level (e.g. global variability), and therefore they can only be addressed with the understanding of the overall emerging behaviour.

The systemic modelling approach accounts for the behavioural description of the infrastructure. It is also associated to the paradigm of control. Controls here refer to the measures that prevent hazards, detect and react to their occurrence and recover the system back to a healthy state. The control is not only the ability of detecting abrupt changes due to local malfunctioning, but also understand the slow degradation of service at the level of SoS [Lev95, Hol06]. This aspect qualifies the control process – from prevention to recovery – under the system attribute **resilience**. Resilience is the key-attribute to understand the behaviour of complex infrastructures. The US DHS [DoHS06] defines resilience as "the capability of an asset, system, or network to maintain its function during or to recover from a terrorist attack or other incident". When service is the focus, resilience is defined as the ability to endure all hazards (i.e. survive and recover) while guaranteeing an acceptable level of service, from which the concept of graceful degradation. In simple words, if reliability returns the limit of a system in withstanding faults, as identified by its structural representation, resilience has to cope with the consequence of the same faults (and more hazards) and control the resulting propagation effects in order to survive and recover.

The systemic approach, the control paradigm and resilience, all together provide a complete theoretical umbrella for the modelling of critical networked infrastructures. In this paper we define a modelling language for critical infrastructures inspired by this systemic approach, and focused on resilience. The scope of the proposal is to develop a methodology for describing complex systems that may present emergent behaviour, and which must embed resilience as system property.

With this objective in mind, we examine the requirements that a language for infrastructures should meet and starting from these requirements, a first version of the Infrastructure Resilience-Oriented Modelling Language (or I®ML) is presented. The I®ML is a domain and application independent language, with a rich ad-hoc semantic oriented to the representation of (i) infrastructures, and (ii) the (inter)dependencies among their components. The semantic follows the control paradigm and provides ad-hoc elements for the representation of control relationships. The process of model building is modular and intuitive, leaving to the analyst the possibility of exploring each single element at the desired level of detail.

The paper is structured as follows. In section 2, the general requirements to model networked infrastructures are given and translated into requirements of the modelling language. In section 3 the language elements are presented with syntax and semantic. Section 4 introduces some elements of analysis with the derivation of the dependency structure from the I®ML model. Section 5 concludes the report with a few remarks.

2 Modelling Critical Infrastructures

This section identifies the goals and the features of the language needed for modelling critical networked infrastructures. Subsection 2.1 presents an overview of the existing methodologies for modelling complex systems; subsection 2.2 discusses goals and requirements and, finally, subsection 2.3 presents the features that a modelling language should possess.

2.1 An overview of the State of the Art

A model is an abstraction of the universe of discourse in which a problem occurs [And04]. Models are used to define either technical artefacts or natural phenomena under limitations that respond to a balance between completeness, i.e. to which extent the model covers the reality, and tractability, i.e. the possibility of handling the problem and return a solution. In the case of complex systems such as networked infrastructures and system-of-systems, this modelling practice encounters serious obstacles. Completeness, in particular, is incompatible with the multiplicity of views that characterize the description of the single parts of the infrastructure, including the number of connections that make them functioning together [Pru07]. Tracking changes, which are common in networked infrastructures, is also another issue. A higher-level view is needed in order to avoid the "forest vs. trees" dilemma and, at the same time, keep an eye on those attributes that emerge only by a systemic view, such as resilience and safety. This is also an indication that one must pay special attention to the points of interaction between different systems (and subsystem) [Per07]. Once interaction points are identified and understood, it is possible to account for the vulnerabilities [Ega07] and the measures necessary to improve the overall resilience.

The definition of a language for the modelling of complex networked infrastructures needs the identification of goals and then the features that make it possible the achievement of the goals. The work of [DeL04] identifies three general modelling aspects: (1) all involved parties to find an answer to their questions, (2) all the relevant issues of the problem to be addressed, and (3) this should be done under a holistic perspective. These three aspects can be used to benchmark the existing modelling approaches. First of all, modelling languages that return the description of heterogeneous networked systems, in a holistic perspective, are not available yet [Ega07]. Instead, modelling approaches focus on specific sectors and specific issues with poor or no consideration on cross-sector interrelationships. They also give more emphasis to the numerical aspects and define rigid and artificial system boundaries without a real-world counterpart [Pru07]. Another limitation stands in the fact that they are not able to balance conflicting goals at the infrastructure level [Fri07], which is one of the main challenges to modelling infrastructures. A networked infrastructure is the aggregation of parts without a preconceived rational designed structure behind and without any single operator or controller overseeing it. Assuming that each part of the networked infrastructure can be assessed within a suitable modelling framework, those same modelling frameworks fail to return reliable results when applied to the overall infrastructure. Said in a different way, the behaviour of the infrastructure and its attributes cannot be inferred as the sum of the behaviour and attributes of its parts – apart from very simple systems-of-systems [Hol06].

The research in this subject has especially found applications in the field of complex socio-technical organizations, while in traditional engineering it encounters less favours and some resistance to deviate from the consolidate practice exists. In this respect, an interesting analogy can be found in the field of economics, where a distinction exists among economic models and econometric models [Del04]. Economic models are explanatory and qualitative, whereas econometric models are applied for (statistical) quantitative analysis of economic factors. As pointed out by De Laurentis [DeL04], both models are needed but, for infrastructure modelling, econometric-like models have been dominating so far. Engineering gives an excessive emphasis to the numerical aspects and privileges the

decomposition of problems in small parts that can be treated analytically, which is correct, but misses the global picture. This cultural background impacts on possible directions to infrastructure modelling. Two different options exist: integrative models (closer to the holistic approach) and specialized models (closer to a reductionist approach). Choosing one or the other direction has implications for the uses of the model and its explanatory capability.

- Integrative models: the integrative models return a complete representation of interactions and communications among the various parts in the networked infrastructure architecture. The scope is to the overall infrastructure. The focus is on the structure, and the properties related to its architecture, i.e. "the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution" [ANS00]. For instance, an integrative model of energy and distribution would describe the service providers with their connections to the customers, distribution networks, operations, bulk generation, electric storages, etc.
- Specialized models: the specialized models are useful for studying the behaviour (by simulation or analytically) of specific phenomena that are subject to well-known patterns. These models are very detailed in order to account for the dynamics that link the different quantities involved.

The majority of modelling approaches to infrastructure are of specialized nature [Pru07]. Nonetheless a few examples of integrative models for infrastructures exist. For instance, the recent NIST draft on Smart Grid Cyber Security [NIST09] presents a "conceptual reference diagram" of the Smart Grid framework based, however, on an ad-hoc notation (that by its ad-hoc nature will be different to the notations used by other institutions interested in this, or other CI). More examples of ad-hoc approaches can be found in [Bai09] (based on a hypergraph-based notation) and [Con06] (based on a consequences network). In particular sectors, traditional modelling tools are used, like network architecture models, but they are unable to identify the issues that emerge when different systems are put together. For instance this is the case of security issues in the electricity distribution [Joh09].

The literature review on this subject can continue, though the goal of this document is not that of being exhaustive, and more examples could be given to show limitations of the tools available, which are not conceived for being used in a comprehensive-integrative manner.

From a methodological point of view, the problem of infrastructure modelling could be seen as a problem of Systems Engineering (SE), but not without cost. In the SE approach, a system is represented under diverse perspectives, either architectural or behavioural, and these diverse views are arranged in a hierarchy, from the high level representation down to the elementary components. The Unified Modelling Language (UML) of the Object Management Group [UML10] is a successful attempt into the direction of organizing this diverse information by a number of specialized models. UML is a general-purpose language and other languages are derived in order to satisfy more specific needs. SysML is the extension of UML for modelling of complex systems [Hau06]. Like UML, SysML provides models (called diagrams), for the system description under diverse perspectives: behavioural, structural, parametric oriented and requirements oriented. The SysML seems a good candidate to satisfy the goals of an infrastructure modelling language. At least, it recalls a few principles of good engineering, for instance in the way complexity is hierarchically organized. Nonetheless, there are aspects that do not match exactly with the goals and needs of networked infrastructure modelling. First of all, it is important to bear in mind that the models considered in SysML are intended to support the system design, while an infrastructure, which is not as a unique system-entity, does not respond to a pre-conceived design. This inadequacy is even more evident when the scope of the representation is the "negative behaviour". The descriptions of the failure and propagations of degraded behaviour and the consequences are not provided by any of the SysML views of the system. In term of risk there are even more limitations within a SysML framework. For

example, the assumption that the operation scenario is part of the boundary conditions and cannot change is unrealistic for the networked infrastructure environment, operation conditions and configuration may change in response to the external demand.

This overview of the state of the art in infrastructure and Systems of Systems modelling already tells us something important. The most needed directions of research will have to cope with the representation of the infrastructure when facing accident scenarios. The aspects of the infrastructure related to risks and their control will be more interesting to analyze than the nominal behaviour. As a consequence, the instruments required (modelling and analysis) would be those capable to describe the system features and the attributes that characterize the infrastructure when this is challenged to survive, with the resilience on top of the list.

2.2 Goals, needs and requirements

This section contains needs and requirements of the modelling language for complex critical networked infrastructures, with a focus on the features that it should possess to model resilience attributes.

Needs and requirements are identified starting from a set of goals. We identify three goals for the modelling language. The goals are necessary to cope with the description of the infrastructures at an integrative level:

- **G1) To provide a comprehensive view**. The language should provide a bird's eye view of a networked infrastructure and their interconnections. This is useful to further scope and prioritize more detailed investigations, without hindering the tractability of the model itself. The view should not be constrained to a single focus. Models will be more helpful if different viewpoints are provided. We emphasize the importance of having structural and behavioural viewpoints with technical and organizational perspectives. Indeed modern infrastructures are technological in nature, but a focus on technology alone is not enough to give a complete view. Infrastructures also contain organizational elements consisting of people which are also prone to (organizational) failure and, if do not perform as expected, could compromise the whole infrastructure [Per07].
- **G2) To model the infrastructure interdependencies**. In safety engineering it is well known that the interfaces among elements are the most vulnerable parts [Per99]. In a similar way, the interdependencies represent for the (networked) infrastructures the physical means through which failure events or perturbations propagate. The modelling language should be able to represent those interdependencies. In so doing, it will facilitate the discovery of vulnerabilities and related cause-effect-consequence chains, as well as the identification of previously unnoticed dependencies among the elements involved.
- **G3) To support resilience-oriented analysis.** A modelling language for the description of infrastructures should contain the elements for describing the resilience-oriented features. A prominent aspect of resilience [Mad09] stands in the ability 1) to avoid and absorb disturbances that affect the system and its operational environment, and 2) to perform dynamic reconfiguration of the systems and restoration to the normal conditions. The resilience features are implemented by mechanisms that may be found at technical, systemic, or organisational levels. The description of these mechanisms is essential to the identification of the interdependencies and in particular the failure and degraded performance scenarios, thus complementing the outcomes of classic accident analysis based on the components (like FMEA [Sto96]).

2.3 Modelling features

The Infrastructure (Resilience-Oriented) Modelling Language (from here on, I®ML) should be able to deal with the representations of the physical, abstract and organisational points of view that are present in any infrastructure.
The physical point of view deals with the representation of the physical objects and their connections; the abstract point of view represents the services, functions and states; the organizational point of view represents the human part and includes procedures, policies, etc. The modelling language should provide different **elements** that allow the modeller to freely consider these three points of view, without further restrictions. The **composition** of those elements into subsystems and systems, and the representation of the internal and external connections intra- and inter- systems are very important and will also form part of the modelling language.

2.3.1 Element-modelling features

We take into consideration three types of elements for the modelling language: domains, resources, and services. Domains are defined according to Jackson's Problem Frames approach:

> *A "**domain**" is a set of phenomena that is usefully treated and represented as a unit in problem analysis* [Jack00].

From this definition, a domain can consider the physical, abstract or organizational points of view, depending on the intentions of the modeller and the level of detail needed. For instance, possible domains are "medical staff" (organizational), "gate" (physical), "gate control" (abstract), etc.

> *A "**resource**" represents any quantity (i.e. goods, material) that can be stored and consumed.*

Resources can be in a state that ranges from full to empty (available or not available, depending on each particular case).

> *A "**service**" is the expected result of a set of interconnected, collaborating domains (possibly including resources).*

The collaboration among domains will be expressed by delimiting those domains within a boundary. What is inside the boundary defines an individual system:

> A *"**system**" set of tightly coupled domains that collaborate together in order to provide a service.*

In this way, a system in I®ML can be considered as a hybrid representation that contains structural (domains and connections), functional (services provided) and behavioural elements (control relationships).

From the identification of the modelling elements, the following features for the modelling language are collected:

- **F1**: I®ML shall provide the means to represent domains.
- **F2**: I®ML shall provide the means to represent resources.
- **F3**: I®ML shall provide the means to represent the services provided by a collaborating set of domains, i.e. a system.

2.3.2 Composability modelling features

The composability features of the language deal with the connections among the elements. According to the element triplet domain-resource-service, three interconnections exist: **domain-domain**, **domain-resource**, and **service-service**. There is no need to express the resource-resource connection, as there are no direct connections among resources (if there are connections, they are indirect, mediated by some other domain in charge of moving materials from one resource to the other or similar operations). The composition of connections, for instance among domains and resources, has already be defined as **system boundary**, and every system so delimited provides a service. The different types of connections will be discussed in turn below:

> a) ***Domain-domain****: A domain-domain is a connection of two domains by means of a control relationship that ties them up.*

This means that the connection is alive and operative, supports the exchange of information[2] among the elements, and pursues a particular goal. The existence of a goal, with exchange of information and control mechanisms inspires to the control paradigm in the classical Systems theory [Ash56]. This aims at describing the connection in terms of goal, goal-effectors and sensors, and (when applicable) of a controller with knowledge of the controlled elements and/or its operation environment. In the simplest scenarios a domain may passively read or write information on another domain. We will consider this case as a particular case of a control loop; an open loop without feedback.

> b) ***Domain-resource****: A domain-resource connection is a connection of a domain with a resource*

These connections are useful to express how a domain provides materials that are stored in a particular resource, or how a domain relies on a resource. A domain can be connected to a resource in two ways: 1) the domain feeds the resource or 2) the domain uses the resource.

> c) ***Service-service****: A service-service connection is a connection of one service with another, to express a dependency relationship*.

Sometimes a service can only be provided if another service is working properly. In a complementary view, services can help to achieve other services. As services may depend on other services, the notation should be able to account for it.

There is another connection, albeit of different nature to the three listed above. This is the assignment that links together **systems with services**. Boundaries are used to group the connected domains that constitute a (sub)system. An interconnected subset of domains (including resources) that collaborate together in order to provide a service should be grouped in order to show their mutual collaboration. A connection must exist that assigns the service to the (sub)system that provides it.

The identified composability features relate to the following modelling features:

- **F4**: I®ML shall provide the means to represent control relationships among domains, including goal, effectors, sensors and internal model issues.

[2] Note that in this exchange among domains other materials can also be exchanged. However, we consider that every material exchange will always be associated to some kind of information, in the same way that a "delivery note" (information) is always associated to a delivery of materials (books, clothes, etc.).

- **F5**: I®ML shall provide the means to express the resource production and consumption by connecting domains to resources.
- **F6**: I®ML shall provide the means to represent service usage and provision, by connecting a delimited set of collaborating domains (a system), to a service.
- **F7**: I®ML shall provide the means to express that some services rely on other services.

In conclusion, we have collected 7 features for the Infrastructure (Resilience Oriented) Modelling Language or I®ML. Just for the sake of convenience, the following table collects the seven features together:

Feature	Description
F1	I®ML shall provide the means to represent domains
F2	I®ML shall provide the means to represent resources
F3	I®ML shall provide the means to represent the services provided by a collaborating set of domains, i.e. a system
F4	I®ML shall provide the means to represent control relationships among domains, including goal, effectors, sensors and internal model issues
F5	I®ML shall provide the means to express the resource production and consumption by connecting domains to resources
F6	I®ML shall provide the means to represent service usage and provision, by connecting a delimited set of collaborating domains (a system), to a service
F7	I®ML shall provide the means to express that some services rely on other services

Now, goals and features can be arranged in a matrix in which the '+' sign indicates that a particular feature contributes to achieve a particular goal, as follows:

	F1	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**
G1: To provide a comprehensive view	+	+	+				
G2: To model infrastructure interdependencies			+			+	+
G3: To support resilience-oriented analysis				+	+	+	+

Goal 1 is achieved because if the modeller is able to represent domains, resources and how they build-up a system that provides a service, then it provides a bird's eye that is unconstrained by a particular technology and that does not discriminate different viewpoints (structural, behavioural, organizational, etc.), as the model is neutral with respect to them. Goal 2 is carried out by the capabilities of representing the intra-system collaborating relationships and the connections of the systems with the services and among services themselves. Goal 3 deals with resilience-oriented analysis and needs a more detailed explanation, as follows.

To build-in resilience, we need to take into account the ability of any resilient system to avoid/absorb disturbances and, later, restore to normal conditions. According to F4, there are control relationships among domains, including details of their effectors, sensors and internal models that keep the status of the controlled part. Exploration of possible malfunctioning in the effectors, sensors or deviations in the internal model (with regard to the actual state) will lead to find undesirable issues that lead to degradation scenarios, affecting the control goal that is trying to be achieved (or maintained). A similar analysis can be performed in relation to resources (F5): lack-of resource, unavailability, etc. In addition, for any system and for each issue found in its control loops, its impact on the services provided by that system can be traced back (according to F6). In this way, we identify different

resilience-oriented issues that could be further prioritized and, or, classified (some of them will be acceptable, or lead to a degraded service or even to a non-admissible state).

Finally, according to F7, a system depends also on externally provided services. Those services should be also accounted in the analysis of failures and consequences.

3 The Infrastructure Modelling Language (I®ML)

This section presents the elements of the proposed Infrastructure Modelling Language (I®ML) with the control paradigm, syntax and semantic of the language.

A case study will be used in order to clarify the language elements and their application to realistic scenarios. The case study deals with an on-scene emergency rescue service in a city. The service relies on prompt ambulance delivery from the hospital to the place where the accident (or any similar event) took place. A radio device links the medical personnel in the ambulance to the hospital. The ambulance management system takes care of everything related to ambulance delivery, receiving the emergency calls and prioritizing them. It is also assumed that an automated traffic control system manages the traffic lights and monitors the status of roads, tunnels, etc. in order to minimize stops and delays in traffic, thus helping ambulances to reach the place of an accident and back to the hospital. The heterogeneity of the components involved and their relationships-interconnections make this to be a valid example of system of systems.

3.1 A few insights on the control paradigm and I®ML

Features F1, F2 and F3 concern the building blocks for infrastructure modelling (i.e. domains, resources, services), while features from F4 to F7 express the conditions on the composability of the building blocks. With the latter features F4-F7, I®ML introduced the basic elements of a control paradigm.

The chosen control paradigm takes into account hierarchical and heterarchical aspects of controls. The relationship between the two “controller-controlled” elements is of hierarchical nature: controls are performed by controllers which are higher-level elements with respect to the controlled, lower-level, elements. One example of control hierarchy is provided by the STAMP accident model [Lev04]. The recent NIST draft on Smart Grid Cyber Security [NIST09] also relies on a hierarchical eight-layer stack, known as the GridWise Architecture Council (GWAC). These modelling frameworks emphasize the hierarchical aspects, which are simpler to identify – and are often in one-to-one relationship with the architecture. Reality is more complex and there are different elements (or groups of elements) that exert complex control relationships over other elements while, at the same time, being subject to control in a heterarchical fashion. A heterarchical paradigm will make it possible to represent control relationships wherever they are needed, without giving importance to the hierarchy. In particular, this is the case of control relationships that are established at the level of system of systems, therefore not designed on purpose, but consequences of interconnections. The I®ML will also include heterarchical controls.

The control paradigm also enables the assignment of goals associated to the control relationship between two elements. The goal and the roles (responsibilities) will be specified by I®ML. The language will give less emphasis to the input-output relationships among elements, shifting that focus towards their dependencies, as they affect the achievement of the control goals. This feature of the language is useful to represent the dependencies that are encountered in networked infrastructures

under a diverse perspective: the high level semantic (i.e. the goal of the controller) is bound to the lower levels of the controlled elements.

3.2 Syntax and semantics

I®ML is a graphical modelling language and its elements are graphical objects that represent domains, resources and services, plus connection elements.

3.2.1 Basic elements

The basic language elements are domains and resources.

Domains are represented with square-boxes, see **Figure 1**. Examples of domains are, for instance, "traffic lights", which represent the actual traffic lights in a city, as shown in the figure.

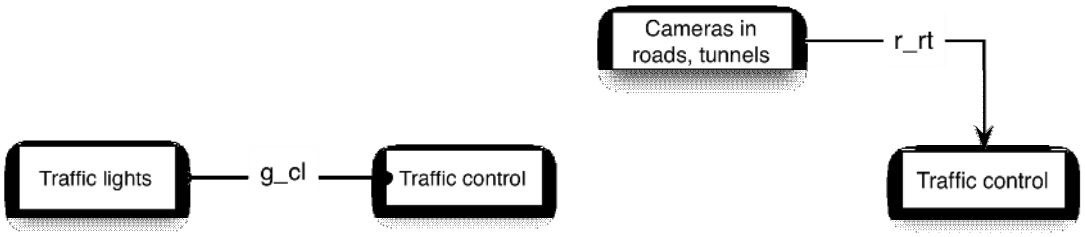


Figure 1. Domains connected. In the left side, the line expresses a control loop with "traffic lights" being the controlled domain. In the right side, a simple arrow expresses that the "Traffic control" domain reads (or accesses to) information provided by the cameras, but there is no exchange of data among them.

Resources are represented with cylinder-shaped elements, like in **Figure 2**, where the ambulance park available is represented as the resource "Ambulances". Domains and resources can be connected together, as shown in Figure 2. The arc that connects a domain to resource storage represents either the addition, the elimination or the update of elements contained in that storage.

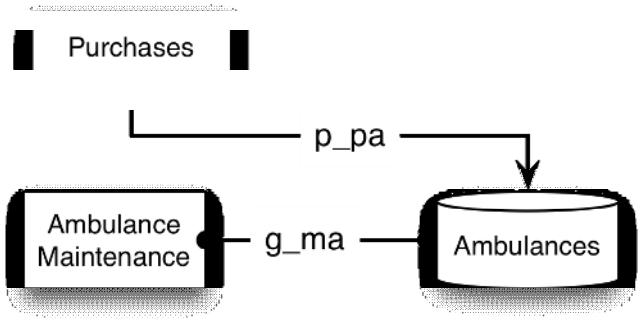


Figure 2. A resource connected to two domains. The lines express the kind of relationships that is taking place ("Purchases" provides the resource and "Ambulance Maintenance" manages it).

3.2.2 Composite elements

Composite elements are derived from the connection of basic elements according to certain rules. In case of control relationship, additional modelling features are specified.

The collaboration of domains, in the simplest case of Domain-Domain connections, is expressed as a control relationship in I®ML. An arc represents the control relationship, with a filled "o" in the controller side and an empty "o" in the controlled side. The basic attributes of a control loop are:

- The control goal: a control system aims at maintaining the goal, e.g. a "set point".
- The effectors: the means to act on the controlled domain. They may be physical devices but also other measures performed by diverse actors.
- The sensors: the means to discern the current state of the controlled domain. Sensors are not only physical devices but also other measures (e.g. job reports, incident reports, after-action reviews, etc).
- The internal model: the controller domain has a (formal or informal) model of the controlled domain. For example a thermostat needs to know the status and dynamics of the controlled environment in response to the applied controls.

Figure 1 already showed a control loop between two domains. A label on the arrow can be used as a legend in order to provide more information about the attributes of the control loop. **Figure 3** shows how an arrow that links two domains, but points in only one direction, denotes a trivial read/write of information from one domain to another. For example, in traffic monitoring, there are cameras that read the state of the roads. Figure 3 shows another example related to the “Ambulance maintenance” domain, that exerts control on the ambulance park (the “Ambulances” resource) at any given moment. The figure also shows how to include control attributes, namely, its goals, sensors, effectors and internal model.

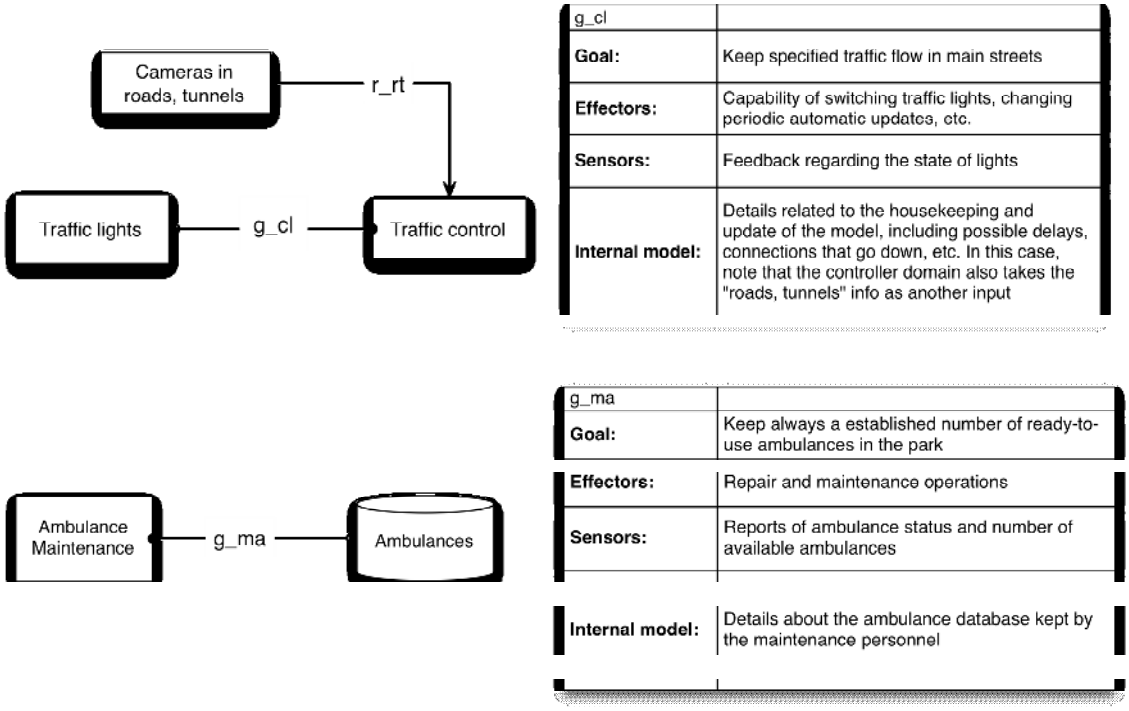
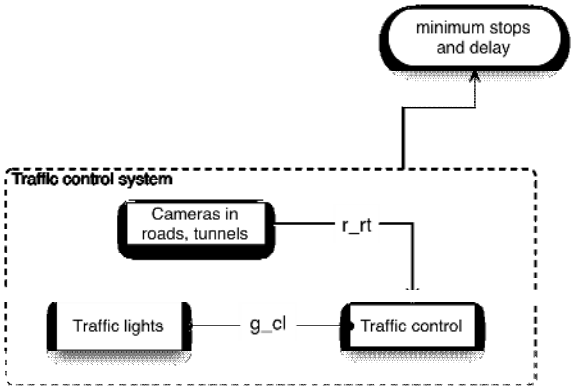


Figure 3: Examples of control loops. Above, the loop labelled "g_cl" links together the domains of "Traffic control" and "Traffic lights". The status of the domain "Cameras in roads, tunnels" is read (but not updated) by "Traffic control", so the relation among them cannot be a control loop. In the right, there are some attributes that further describe the control goal of g_cl, including the effectors, sensors and internal model details. In the example below, Ambulance Maintenance exerts control over a pool of ambulances via the loop g_ma. Again, at the right side, its control attributes are shown.

Systems are made of composite elements, defined as a set of collaborating domains that, together, provide a service. A system is graphically represented by a dotted line that encloses the domains, including their connections, like in **Figure 4**. In this figure, there is a domain, "traffic control", in charge of controlling the "traffic lights" domain (more details can be provided by describing the attributes of this control relationship like sensors, actuators, etc.). The "traffic control" domain has access (reads) the information coming from the "cameras in roads and tunnels" domain. Basically, "traffic control", based on the information provided by the camera, takes decisions regarding the behaviour of the traffic lights. The three domains work together to carry out the "minimum stops and delays" service. They constitute the system, the boundaries of which are expressed by the dotted line. The service is assigned to the system and it is represented with an oval. Additionally, each service can be further specified with a table like the one in the right side of Figure 4, with the conditions under which it is considered as acceptable, degraded or unacceptable. These degrees of service quality depend on the definition of the service and they are up to the modeller or the domain of application (other qualitative or quantitative schemas are possible for qualifying the services). Additionally, the figure represents an important feature of systems in I®ML: the boundary of responsibilities. Every system is responsible of providing the service and every service should be provided by a system. In a sense, assigning a service to a system establishes not only a responsibility, with regard to that system, but a relationship of independence of the service with regard to the other systems.



Service levels	Service name: Minimum stops and delays
Acceptable:	*[description of what is considered nominal behaviour for this service]*
Degraded:	*[description of what is considered degraded, but still functional, behaviour]*
Unacceptable:	*[description of what is considered unacceptable behaviour]*

Figure 4. A set of collaborating domains constitutes a system. Systems provide services; in this case, the service is "Minimum stops and delays". For each service a separate table should specify what is considered an acceptable, degraded or unacceptable level.

A service may rely on another service, which is another connection. An example is shown in **Figure 5**.

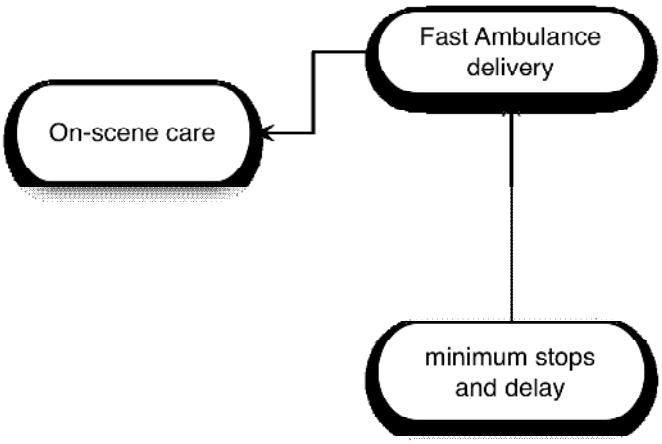


Figure 5. Services rely on other services in order to properly succeed. If "minimum stops and delays" cannot be achieved, or is "degraded", then "fast ambulance delivery" and "on-scene care" will be compromised. For convenience, the figure does not show the subsystems that provide the services.

It can be argued that there are only two levels in this representation: domain and system. However, that should not pose a problem for the modeller. Generally speaking, there are three basic modelling primitives, namely, partition, abstraction and projection [Dav89]. Partition relates to the aggregation ("part-of") relationship. Abstraction deals with the generalization ("is-a") relationship. Projection separates a model into manageable parts, so the modeller can focus on different "projections" or different sections at a time. Different modelling languages combine those primitives in very different ways and with different emphasis. For instance, STAMP does not use abstraction at all (in the "is-a" sense) and Entity/Relationship database models rely almost in projection. In the case of I®ML there is no abstraction (again, in the sense of "is-a" relationships), partition has two levels (a domain is "part-of" a system) but projection is extensively used, allowing the modeller to focus on different pieces (systems and services) of the model but, at the same time, raising the status of the dependencies among those pieces (expressed through service and goal dependencies).

The seven modelling features are shown in the following table with the language graphical elements that represent them in the I®ML.

Feature	Description	Notation
F01	I®ML provides the elements to represent domains	Square
F02	I®ML provides the elements to represent resources	Cylinder
F03	I®ML shall provide the means to represent the services provided by a collaborating set of domains, i.e. a system	Oval linked to a boundary line that delimits a particular system
F04	I®ML shall provide the means to represent control relationships among domains	Line linking two domains, with filled dot in the controller side and empty dot in the controlled side
F05	I®ML provides the means to express the resource production and consumption by connecting domains to resources	Line linking a domain and a resource
F06	I®ML shall provide the means to represent service usage and provision, by connecting a delimited set of collaborating domains (a system) to a service	Systems as dotted lines enclosing a set of domains. Services are connected to systems.
F07	I®ML shall provide the means to express that some services rely on other services	Arrow from one service to another

3.3 Modelling insights with I®ML

This section will discuss the modelling expressiveness of I®ML when used as tool for the representation of critical networked systems and infrastructures in term of resilience.

An I®ML model includes the representation of the services provided by some systems to an external user, and of the control goals that must be achieved for the delivery of those services. The concepts of services and goals, though closely related to each other, present unique distinctive features. If services are provided by systems, and systems have internal (control) goals to be satisfied, the provision of the service relies on the successful achievement of those goals. On the other hand, if one of those internal control goals is not achieved, the service may be compromised to a certain extent, which depends very much on the system resilience features – i.e. the ability to deals with degraded behaviour scenarios. As a system may rely on the provision of an external service, it is possible to identify a network of service-goals relationships. Again, the fact that modelling is focused on control relationships broadens the number of identifiable relationships.
The emergency rescue case study is an example of network of systems, the services of which are interdependent. In Figure 5, the service “fast ambulance delivery” relies on the “minimum traffic stop and delays” service (e.g. provided by a traffic control system). In its turn, the “fast ambulance delivery” contributes to the “emergency support” service. In this way we can express the cooperation among different systems and, or, other systems that provide the services. The same description supports the analysis of accident and degraded behaviour scenarios that occur locally and propagate throughout the network. This is the topic of next section 4.

The modeller has different options for choosing the domains to be represented in a model. Those options help to shift the focus to different issues and facilitate the modelling of heterogeneous operation scenarios. Domains can be a mix of people and technology. For instance, in the case study, the domain named “Assignments” represents that someone is in charge of assigning ambulances to drivers and medical personnel. So the "Assignments" domain describes a real department or group of people (with supporting technologies) that are in charge of assigning ambulances. That department, represented by the "Assignments" domain, has a controller relationship with other controlled domains, namely “Ambulances” and “Personnel”, otherwise it would not be able to perform the assignments properly, achieving its goal.

Resources will make it possible to answer questions related to the lack/surplus of a particular good. Malfunctioning during operation time may be discovered by considering what happens if the resource is disconnected from their neighbouring domains, or if they fall below or above an established threshold. The implications of those situations must be taken into account, including their impact on the services provided by the system where the resource is located.

3.4 Guidelines for the construction of I®ML models

This section introduces the "how-to" aspects of model building. It is important to remark that there is no “best way” to start building a model, and each case studied will be different, depending on the purpose, the perspectives of the developers and the needs of the users. The proposed approach is service-driven, therefore top-down. This fits well anytime the purpose is to build a descriptive model. The approach consists of the following steps, applied in sequence:

- Identify and enumerate the services involved with their links and dependencies.
- If a service depends only on another service(s), link them with an arc with an arrow oriented into the direction of the dependency.

- For each service that does not depend directly on other services, list the domains that are needed in order to provide the service.
- For each domain try to identify the control relationships by addressing the following questions: Is this a controller or a controlled domain? If it is a controller, which is the controlled domain? And, which is the goal that controller and controlled are trying to achieve? In this way we obtain pairs of domains in a controller-controlled relationship, taking also into account that a controller domain in a relationship can be also a controlled one, via a different relationship with a third domain. Once the different domains are connected via controller-controlled relationships, we can draw boundary lines that define the (sub)systems that provide the services identified. In this way, we have defined each system by interconnecting and delimiting the domains that collaborate together.
- Once the systems are identified, for each of them we ask: does this system rely on any externally provided service(s)? If the answer is yes, the system will be connected as dependant on the new service(s) found, and those services will trigger a further search of new domains and systems, possibly dependant on other services, etc. This process is repeated again at the discretion of the modeller who, to avoid infinite regression, will decide when to stop whenever he or she considers that some services are out-of-scope of the analysis.

Following the process for our example, in step 1, service identification, we arrive at "on-scene care", "fast ambulance delivery", "reliable radio contact" and "minimum stops and delays". In step 2, dependency arcs are traced that link "on-scene care" as directly dependant on "fast ambulance delivery" and "reliable radio contact". In addition, "fast ambulance delivery" depends on "minimum stops and delays". All these services identified do not depend on other services, with the exception of "on-scene care". Hence, in step 3, we list the domains that are needed to provide those services. In this way, we arrive at the conclusions that:

- "Minimum stops and delays" needs to be provided by a collaboration of "traffic control", "traffic lights" and "cameras in roads and tunnels".
- "Fast ambulance delivery" needs a complex collaboration among domains such as "ambulance maintenance", "assignments" of ambulances to personnel, "purchases" of new ambulances according to "demand analysis" that relies on "historical data", the actual park of "ambulances", "contacting" of "personnel", and "budget". In this list, "budget", "historical data", "personnel" and "ambulances" can be treated as resources.
- Other collaborating domains provide “Reliable radio contact”, but we will not analyze them in this example to keep it manageable.

Once those domains were identified, in step 4 we need to establish control-controlled relationships among them. In this way, we find that "assignments" controls the "personnel" and "ambulance" domains, or that "traffic control" exerts its control upon the "traffic lights" domains, etc. Once the connections are found, the lines delimiting each system are drawn. In this case, we arrive to two systems named "ambulance management system" and "traffic control system", so we can proceed to step 5. In this step, we find that the "ambulance management system" depends on the correct functioning of two services that will be called "incident information" and "budget policy". The first service gives raise to a new system composed by newly identified domains ("Incident management system" composed by the "incident recollection", "incident prioritization" and "incident data" domains). This system depends on the correct functioning of "Emergency calls 112" service, that, together with the other service ("budged policy") are left out of the scope of our analysis. On another side, we find that the "traffic control system" depends on a "camera maintenance" service but we will not further analyze this issue.

The result of this process is in **Figure 6.** The main "on-scene care" service relies on the "fast ambulance delivery" service. The "fast ambulance delivery" service is provided by the "ambulance management system" that consists of a set of collaborating domains like "assignments" of personnel and ambulances, "ambulance maintenance" that takes care of the ambulance park status, "demand analysis" that relies on historical data to provide an adequate number of ambulances and personnel, etc. The "ambulance mgmt. system" has external dependencies on two services: "budget policy" and "incident information". This last service is provided by another system, named "incident mgmt. system" in the picture, which relies on "emergency calls" to provide an adequate prioritization and localization of reported incidents. On another side, "minimum stops and delays" in highway traffic strongly influences "fast ambulance delivery". This "minimum stops and delays" service is provided by the traffic lights controlled by the "traffic control" system, a system which, among other things, relies on properly maintained cameras, as expressed by the dependency on the "camera maintenance" service. In a similar way, the "on-scene care" service relies on "reliable radio contact" (this last item and its dependencies are not explored in this example).

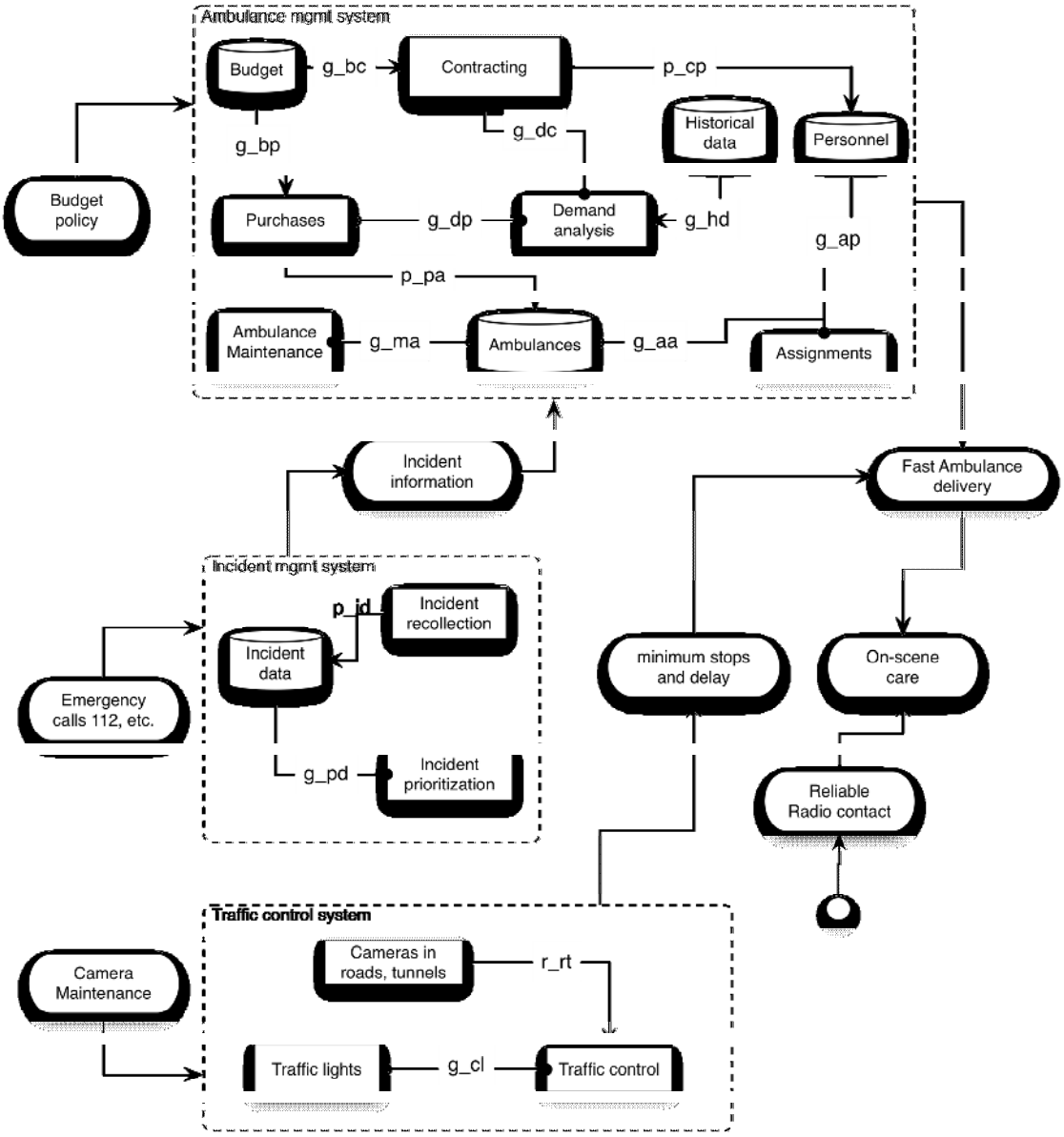


Figure 6. Example of the infrastructures related to on-scene care.

I®ML models lend themselves to further investigation of dependencies, propagation of failures and expected service degradations. This topic will be tackled in the next section.

4 I®ML-based Analysis

The last section introduced the "how-to" aspects of model building. In this section we will provide a brief overview of the possible uses of the model for analysis. An I®ML model is suitable to represent those scenarios that challenge the ability of the systems to survive to undesired events, therefore showing its resilience and the effectiveness of the resilience features. To this end, it is important to derive from the I®ML model the structure that renders possible the traceability of the events that progressively lead to a disservice. This representation format will be called “dependency structure”. The outcome of this dependency structure can then be used to devise adequate measures that counteract the failure or help to recover the system in a resilient fashion, like the protection measures applied to reduce the risk.

4.1 Dependency Structure

A "dependency structure" rearranges the information in the I®ML model in order to highlight the (inter)dependencies among services, systems and control goals at domain level. An example of dependency structure is shown in **Figure 7**, which refers to the I®ML model of Figure 6. In Figure 7, the service “On-scene care” is shown as dependant on two services: “Fast ambulance delivery” and “Reliable radio contact”. The first one is influenced by “Minimum stops and delays”, which is a service provided by the “Traffic control system” and, henceforth, depends on achievement of the internal control goals of that system (the “Traffic control goals”) which are the two goals of the two controls loops internal to the “Traffic control system”, namely “Keep specified traffic flow” and “Cameras status”. This last one depends on the achievement of “Camera maintenance”. The remainder of Figure 7 can be described in a similar way.

The dependency structure is a tool to (i) represent the dependencies among the different goals and services involved in a complex networked infrastructure and (ii) a support to analyze the propagations and repercussions of undesirable events (e.g. mishaps, faults, disruption of service) that concatenate across the different elements. The expression “dependency structure” comes from the first usage of the structure, but “propagation structure” would also be a perfect alternative name.

The modelling language forces to think in terms of resilient mechanisms and control relationships, and this fact has valuable implications for the analysis. In particular, the use of control relationships provides insights on the events that accompany the deviation of the system from nominal conditions to anomalous ones. While the traditional instruments available in incident analysis (e.g. safety, security, ...) build the scenarios in a “linear fashion” (e.g. an event triggers other events up to the end states) here more intricate chains of events and states might be identified: those related to the dependencies among systems and services. This information can not always be inferred from the analysis of the single parts, most notably when emergent behaviours are observed [Hol06, Lev95]. This feature of the language is an important leap forward in the understanding of complex networked infrastructures.

The dependency structure, by itself, is not informative, and an exploration is necessary to generate the failure scenarios. A “what-if” reasoning is at the basis of the exploration: the disruption from nominal behaviour is introduced somewhere and its propagation is tracked. An example, taken from Figure 7 is the following: if the goal of "adequate number of personnel in place" is not being achieved, then it will have an impact on "fast ambulance delivery", with negative consequences for carrying out the "on-scene care" service. The way to derive the dependency structure of Figure 7 will be presented in the next subsection.

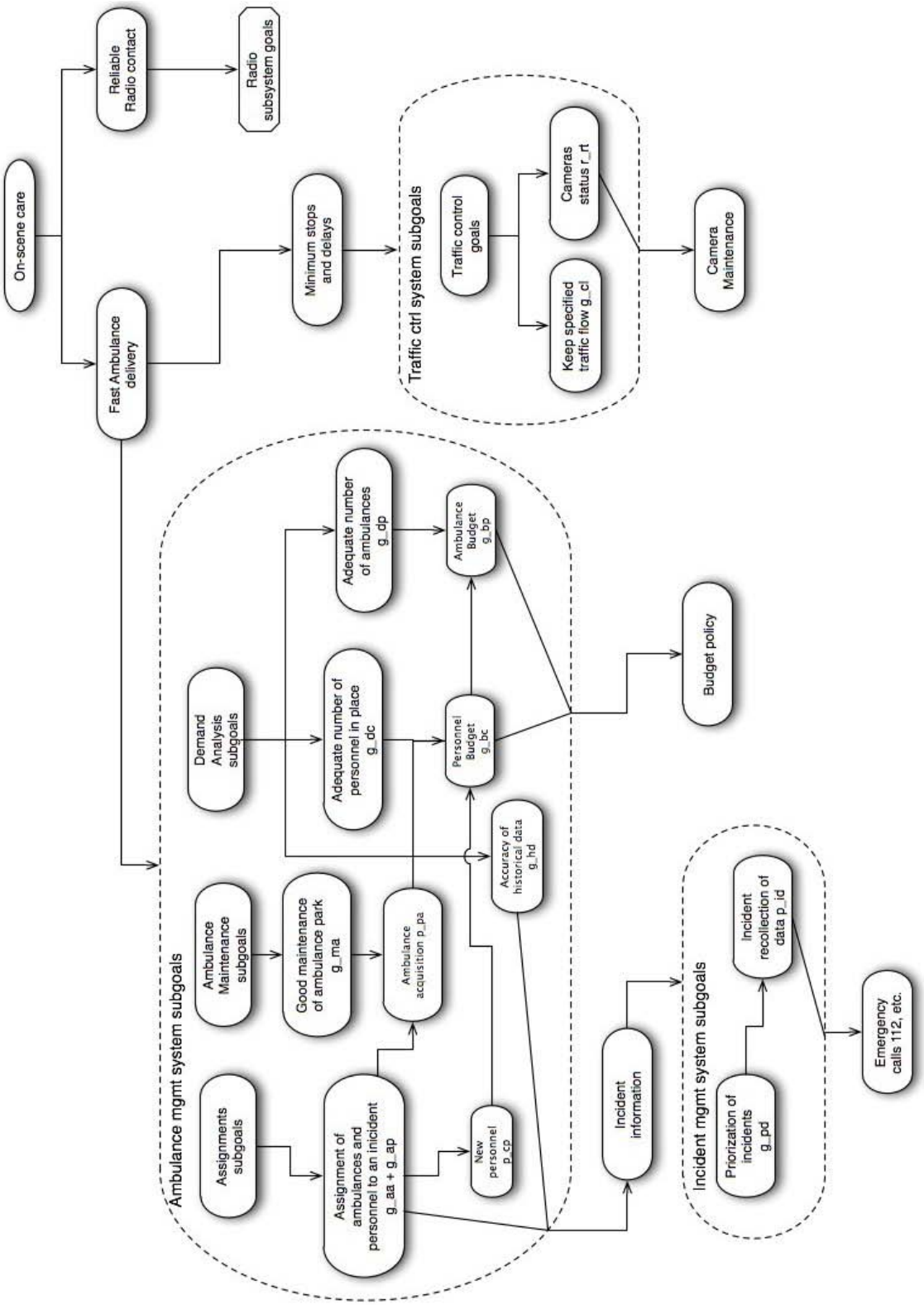


Figure 7. Dependency structure related to the IML diagram from Figure 6.

4.1.1 How to derive a dependency structure

An I®ML model can be easily converted into a dependency structure. For instance, **Figure 8** shows part of an I®ML model accompanied by its derived dependency structure. This dependency structure shows the services provided by a system, but the system itself does not appear in it. All internal components have been replaced by the goals of the control loops they are involved with. The focus of the analysis is on degraded behaviour under failure or other conditions, which is necessary to discover

resilience gaps; so, from Figure 8, it can be deduced that failures and improper performance of the “incident information” service are caused by the non-achievement or improper performance of the “prioritization of incidents” subgoal of the Incident Management System, and the non-achievement of the "prioritization" subgoal could be traced back to “incident collection of data”, which depends on an external service devoted to the gathering of emergency calls and associated data (address, kind of incident, etc.). We can see here how [service-goal-...] chains of dependencies are built in order to drive the analysis.

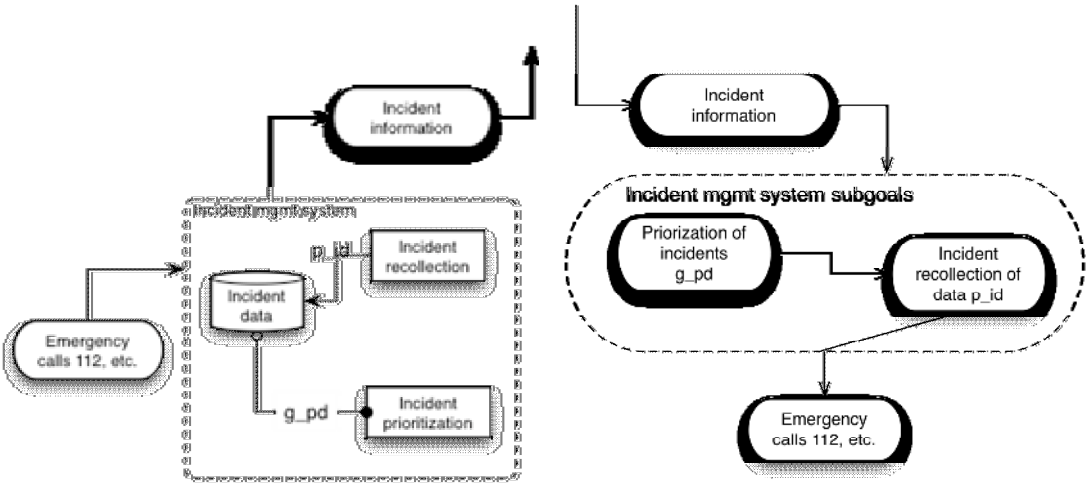


Figure 8. An I®ML diagram (left) and its corresponding dependency structure (right).

So far, we have explained the example in terms of "failure" and "degraded performance" or "improper behaviour", with reference to goals and services. This is because we want to avoid a black-or-white approach (failure vs. correct behaviour), so it can be said that degradations in the performance of a service (like the "incident information" service performing under substandard timeliness) can be traced back to non achievement or partial achievement of goals in the system that provides the service (like lack of performance in the "prioritization of incidents" subgoal, who could be traced back to lack of performance on the "incident collection of data", etc.). The resilient features are meant to face failures and control the degradation of services. Moreover, they also anticipate their occurrence.

The procedure to derive a dependency structure from the I®ML model consists of the following steps.

For each system:

- Within the system, for each arrow that connects two domains, there will be an internal control goal that must be achieved. Those intra-system control goals are drawn as ovals.
- Connect the ovals that represent internal goals. This will be done based on the concept of “shared controlled domains”: whenever there is a domain (including storages) shared by two or more control loops, there is a dependency among the goals of those loops. It can be said that goal A depends on another goal B if the controlled domain in goal A is also the controlled domain in B. For dealing with this situation graphically, draw an oval corresponding to control goal A, another oval corresponding to control goal B and, finally, draw an arrow from the oval of A towards that of B. For instance, in Figure 8, the “prioritization of incidents” goal is connected to “incident recollection of data”, as these two goals are related to the controlled domain “Incident data”.
- Proceed in this way with the goals of all the control loops contained within a system until no more connections are possible. When finished, a dotted-line oval must be drawn around the connected goals found, to express their intra-system nature. In Figure 8 that oval is labelled as “Incident mgmt system subgoals”.

- Connect each service with the dotted-line oval that surrounds the control goals of the system that provides that service. Draw an arrow from the oval representing the service to the dotted-line oval. In Figure 8 this is the arrow that goes from “Incident information” towards “Incident mgmt system subgoals”.
- Connect the services on which the system relies on, with an arrow from the dotted-line oval towards the service. In Figure 8 this is the arrow that goes from “Incident mgmt. system subgoals” to “Emergency calls 112”. However, this connection is just a first approximation, as it does not specify which subgoals within a system are, in particular, dependent on the service, as not all of them are (for instance, in the figure, "ambulance acquisition" does not depend on "incident information"). For this reason, another step is needed
- Within a system that depends on a service, find which of the internal goals of the system are actually related to that service and draw a dotted line form the goal to the service. For example, the lines in Figure 8 that go from "personnel budget" and "ambulance budget" point to the service "budget policy", as they are directly dependant on it.
- Once all systems have been converted into connected control goals, as explained above, the last step is to connect services that depend directly on other services.

The dependency structure is built bottom-up but this does not mean that it should be used, or read, in a bottom-up fashion. It is important to remark that the non-achievement of an intra-system goal corresponds to a failure in the controls (or in one of the external services that the system depends upon). Those malfunctions will have consequences on the services provided. Again, a resilience feature may mitigate these possible consequences, and this intervention may be transparent to the user of the service up to a certain extent, when the performance drops down below a given threshold.

4.2 Analysis and dependency structure

The potentiality of the dependency structure emerges when considering paths of several elements as well as loops. In Figure 8, the impact of improper "camera maintenance" falls onto the traffic control system, so it cannot achieve some of its internal control goals ("camera status", in this case), with consequences on the “minimum stops and delays” service, which in its turn will have an impact on “fast ambulance delivery”, hence leading to undesirable consequences for “on-scene care”, and so forth.

Another insight that can be deduced from Figure 8 is the following: "if incident information is not adequately collected, incidents will not be properly prioritized. As a consequence, the incident information gathered will not be sufficient to do a correct assignment of ambulances/medical personnel to an incident, with consequences to fast ambulance delivery and on-scene care. This suggests that an adequate incident data collection is needed for quality on-scene care".

The dependency structures can be very complex for non-trivial systems, and some recommendations should be available in order to derive and use them. With respect to traditional system analysis techniques (FTA, Even Trees, FMEA, etc.), where the exploration begins "in the void", the analysis with I®ML and in particular the generation of the dependency structure, takes big advantages. The information is in the model, including the control relationships. To this end, and without the intention of being exhaustive on the topic, some analysis insights worthy of consideration are pointed out in this section.

4.2.1 Resilience scenarios

From a comprehensive point of view, every goal and service in a dependency structure can fail or perform under substandard conditions, at least potentially. The structure does not help in finding which

particular goal or service will fail or how (for instance, it does not help to find security vulnerabilities). That's a task that belongs to other techniques. Work with a dependency structure starts by assuming that one or more goals are non-achievable, but the structure itself does not define what are the goals to focus on. Once those goals are defined, the dependency structure comes into play. The links in the structure help to point out how local off normal conditions (using a very neutral term) propagate and influence other goals and services. This has also a value for the designer who is informed on the most critical parts that should be enhanced and the possible measures to take.

The path that starts from the failing goals and lead to the consequences for the service will be called "resilience scenarios". The name wants to recall the ability of the model to cope with off-normal behaviours in a resilient fashion. Those paths do not necessarily lead to a disservice, as said, yet it is important that they are all identified in order to assess resilience. From an analysis point of view, once a plausible resilience scenario is identified, some ideas can be put in place in order to devise measures for prevention, detection, absorption, protection and recovery [Mad09].

4.2.2 From a dependency structure to resilience scenarios

As previously mentioned, a dependency structure only shows the expected behaviour of the infrastructure, and does not provide the failure logic to infer a disservice from the non-achievement of a goal, the unavailability of a resource, etc. Nonetheless, it provides the view of the infrastructure as system of systems, with the necessary anchor points to explore the effect of any undesired event. The following questions may help to conduct the exploration:

i) What are the services potential affected by the non achievement of a goal?
ii) To which extent is the goal not achieved?
iii) For how long does the off normal condition persist?
iv) Do resilience measures exist to absorb/retard the consequences or recover?

The number of questions is necessary to avoid a trivial black-or-white (e.g. failure vs. correct behaviour) classification of a given resilience scenario. Indeed, the resilient measures are in place to face and control the degradation of services, dynamically. Moreover, they also anticipate or retard the occurrence of such degradation acting like as buffer-damping elements.

A “what-if” reasoning (forward-inductive) is therefore at the basis of the generation of resilience scenarios for a given dependency structure. The step zero is about to define the event which may determine the off normal condition somewhere in the structure. This is typically corresponding to the non-achievement of a goal, but as said it can also be the unavailability of a resource. The step 1 refers to point i) in the list: the resilience scenario setting with the chain of dependencies. An example is taken from Figure 7. If the goal of "adequate number of personnel in place" is not achieved, then it will have an impact on "fast ambulance delivery", with negative consequences for carrying out the "on-scene care" service. The second step concerns questions i) and ii). Last step responds to question iv) and completes the definition of the resilience scenario with the assessment of the existing resilience measures.

The direction of exploration of a dependency structure can also be backward or deductive. If the structure is explored backward, from the service malfunctioning back to the goals, the meaning of the connections is "to rely on" (e.g. "on-scene care" relies on "fast ambulance delivery" and also on "reliable radio contact"). A backwards analysis typically supports an accident analysis; the accident is tracked back to the causes. In this case, it is possible to postulate the non-achievement of more goals at a time.

In our intention, the dependency structure is to be considered as a generator of resilience scenarios and it can play the same role of an accident sequence analysis in traditional risk analysis settings. For

example, the resilience scenarios can be used for a risk-informed prioritization of the most critical scenarios. The result of the analysis is a set of accident scenarios to which it may be added the analysis of resilience. This analysis of resilience, if including also performance as a quantity of interest, needs the representation of the transient behaviour from the occurrence of the disruptive event to the recovery of the system (or systems) back to its normal state. The analysis, coupled to the dependency structure, will act in a way similar to accident sequence analysis in traditional risk analysis settings. However, issues related to recovering measures, their design and the trade-offs involved, are out of the scope of this document.

5 Conclusions and Further Work

This report introduced to the I®ML language for modelling and analyzing infrastructures. The decision to develop such a language emerged after observing that there is a lack of standard approaches to infrastructure representation and analysis, i.e. a modelling framework. However, lack of modelling frameworks has already happened in other fields and was appropriately solved. For instance, in Software Engineering, during the 70s and 80s, the description of the architecture of complex software systems was dealt with rather informally (just "boxes and arrows"). It was in the 90s when Software Architecture emerged as a standalone discipline bringing some clarifications and reorganization to the topic and paving the road to future developments [Sha96, Tyr05]. The same theoretical leap forward is envisaged for the novel discipline of critical infrastructures.

The proposed I®ML language is based on the control paradigm and heterarchical relationships among the different elements: domains that collaborate together to build up systems with the purpose of providing a service. The widespread applicability of the control-controlled relationship makes I@ML models able to accommodate different complementary views like physical, abstract or organizational.

In this way, the language makes it possible to identify dependencies. Services may be mutually dependant on each other. One dependency is directly derived through the chain of systems that rely/provide services. Another dependency links services that contribute and help to each other (e.g. the ambulance service depends on traffic routing). This document not only introduced the I®ML language and model construction, but its use in analysis as well, by deriving a Dependency Structure. This structure helps the analyst to focus on the different mutual interdependencies and is the basis for resilience-oriented analysis of CIs.

The language provides a valid support to safety and risk analysis, with potential applications in resilience design, accident forensics, etc. Modelling a given reality by means of I®ML allows analysts to explore the problem space and evaluate the effectiveness of measures in place beforehand. Modelling is a widely used technique in safety engineering, where failure models support the analysis of accidents [Lev04]. These models can be used in a post-mortem fashion (after the occurrence of the event) or preventively, as background to risk assessment, specialized in the technical components, the human behaviour, or both [Hall08]. The dependency structure provides a list of accident (incident) scenarios as result. Each of them may be further analyzed to see, for instance, how resilient are the systems facing particular hazards or disruptions or for decision-making on which measures should be put in place to avoid or ameliorate the malfunctioning discovered. For instance, in the example we have been following in this document, a scenario was discovered where quick ambulance delivery suffered the impact of a lack of camera maintenance in roads and tunnels. From that discovery, different measures can be designed and traded-off with their alternatives, in order to ameliorate the problem. Development of the methods and tools that will allow this kind of detailed analysis, and the consequential design and implementation of possible measures, is in our agenda for further work.

Further research work, despite being based on I®ML, on the one hand, can be framed within the approaches and traditions of risk analysis and safety engineering but, on the other hand, it should take into account the essential differences between "traditional" approaches and resilience-oriented approaches. Enumeration of accident scenarios derived from a dependency structure will be very similar to the enumeration of accident sequences in a risk analysis framework, where each sequence is analysed with probabilistic tools. Additionally, and still in analogy with safety analysis, we can use infrastructure models to find vulnerabilities and dependencies, previously unnoticed, followed by design measures to implement-enforce resilience. In this respect, similarities exist between this proposal and the safety-oriented techniques like HAZOP or FMEA [Kletz97, Redm99]. Analysis in techniques like HAZOP or FMEA starts with a design diagram of the system. Other techniques like FTA or Event Trees [Lev95] are more focused on event sequences that, however, assume the existence of a design diagram as well. In HAZOP, for instance, components and connections are dealt with individually and qualitative guidewords (“more”, “less”, “late”, etc.) are used to facilitate a creative thinking about possible mishaps and vulnerabilities. In FMEA, predefined "failure modes" are investigated for its possible impact on the different system components that appear in the diagram. In our approach we also start with a diagram of an actual infrastructure with their systems and their connections. This has been the main focus of the document. Detailed techniques for finding vulnerabilities and weak points followed by designing possible resilience-oriented measures will be addressed in the future with approaches not dissimilar to the ones discussed above, at least in a first approximation. However, we are aware that bringing ideas from "orthodox" techniques to the context of resilience-oriented infrastructures is limited by different reasons:

- Resilience is a feature that applies to systems developed by "rafting" different systems already in place [Ega07]. Unlike safety, then, resilience is something not built-in on purpose.
- In traditional engineering, a system may fail because of an endogenous fault. However, resilience is very concerned with external sources of disturbances/failures - e.g. dependencies can be exogenous.
- In other fields there is an all-or-nothing approach to failure: a system or component fails, or not. For resilience, it is important to specify that a service can be provided in admissible, degraded and non-admissible levels. Resilient measures should intervene at the approaching/reaching of a degraded and non-admissible state, while not interfering on the normal behaviour.
- It must be taken into account the existence of diverse alternatives for providing the same service (triggered by different environment conditions) or for coping with malfunctioning by activating some emergency measures that could impact on other systems and services.
- Control relationships and resilience measures in the IRML should represent an added value as they may return more insights on a system malfunctioning than a FMEA performed on the static architecture. For example, it is possible to discover scenarios where the triggering event is not a failure but rather the improper application of a resilience measures in one system, which destabilizes other systems in the network.

Other issues that deserve further research work are the following:

- To develop risk/probabilistic semi-automated tools to further analyze the different scenarios that emerge form a dependency structure. Bayesian networks [Ben07] mapped to a dependency structure are a promising approach.
- To provide support for forensic and after-the-fact analysis, in order to find explanations and facilitate the understanding of the different contributing factors and their relative impact.
- To provide explicit support for the analysis related to lack of resources. How could a lack of resource or non-access to it influence on other systems?
- Take into account timeliness of disruptions. For instance, lack of electricity, after some hours, could bring down the mobile phone network, as all mobile batteries will lose their charge.

- Analysis based on particularly disruptive events whose consequences are widespread through the network of systems. Given an I®ML model, different analysis should be performed for particular circumstances like earthquakes, flooding, etc.

In our research plan, there is also the intention of developing the "institutional dimension" of services, control goals and other elements in the models. Given the institutional fragmentation that nowadays pervades actual CIs [DeB07], some tool that helps to make a distinction within complex networks and boundaries of responsibilities will be very useful.

6 References

[And04] J. Andrade, J. Ares, R. Garcia, J. Pazos, S. Rodriguez, and A. Silva, "A methodological framework for viewpoint-oriented conceptual modeling," *IEEE Transactions on Software Engineering,*, vol. 30, no. 5, pp. 282-294, 2004.

[ANS00] ANSI/IEEE 1471-2000, *Recommended Practice for Architecture Description of Software-Intensive Systems.* 2000.

[Bai09] F. Baiardi, C. Telmon, and D. Sgandurra, "Hierarchical, model-based risk management of critical infrastructures," *Reliability Engineering & System Safety*, vol. 94, no. 9, pp. 1403-1415, September 2009.

[Ben07] I. Ben-Gal, "Bayesian Networks". in Ruggeri, Fabrizio; Kennett, Ron S.; Faltin, Frederick W. In *Encyclopedia of Statistics in Quality and Reliability*. John Wiley & Sons. 2007.

[Bom09] E. Bompard, R. Napoli, F. Xue, "Analysis of Structural Vulnerabilities in Power Transmission Grids", *International Journal of Critical Infrastructure Protection*, Elsevier, Vol. 2 pp. 5-12, 2009.

[Cas08] E. Casalichio, E. Galli and S. Tucci, "Modeling and Simulation of Complex Interdependent System: a Federated Agent-based Approach", CRITIS 2008, *Springer Verlag*, pp. 72-83, 2008.

[Con06] S. H. Conrad, R. J. LeClaire, G. P. O'Reilly, and H. Uzunalioglu, "Critical national infrastructure reliability modeling and analysis," *Bell Labs Technical Journal*, vol. 11, no. 3, pp. 57-71, 2006.

[Davis89] A. Davis, *Software Requirements: Analysis and Specification.* Prentice-Hall, 1989.

[DeL04] D. DeLaurentis and R. K. Callaway, "A system-of-systems perspective for public policy decisions," *Review of Policy Research*, vol. 21, no. 6, pp. 829-837, 2004.

[DoHS06] Department of Homeland Security, U.S. *National Infrastructure Protection Plan*, 2006.

[Ega07] M. J. Egan, "Anticipating future vulnerability: Defining characteristics of increasingly critical infrastructure-like systems," *Journal of Contingencies and Crisis Management*, vol. 15, no. 1, pp. 4-17, March 2007

[Fri07] A. Fritzon, K. Ljungkvist, A. Boin, M. Rhinard, "Protecting Europe's critical infrastructures: Problems and prospects," *Journal of Contingencies and Crisis Management*, vol. 15, no. 1, pp. 30-41, March 2007.

[Geo10] A. V. Georghe, M. Masera, "Infranomics: a Discipline of Discipline for the XXI Century", International Journal of Critical Infrastructures, *Inderscience Publisher*. To be published in 2010.

[Hai08] Y. Haimes, "Models for risk management of SoS", *Int. J. System of Systems Engineering*, Inderscience Publisher,Vol. 1, pp. 222-236, 2008.

[Hau06] M. Hause, The SysML Modelling Language. *Fifth European Systems Engineering Conference*, 18-20 September 2006.

[Hey89] F. Heylighen, "Self-organization, Emergence and the Architecture Complexity", *1st European Conference on System Science*, Paris, pp. 23-32, 1989.

[Hol06] *Resilience Engineering: Concepts And Precepts*, Erik Hollnagel, David D. Woods, Nancy Leveson (Editors), 2006.

[Joh09] E. Johansson, T. Sommestad, M. Ekstedt "Issues of Cyber Security in SCADA-Systems. On the Importance of Awareness" *20th International Conference on Electricity Distribution.* Prague, 8-11 June 2009.

[Joh06] C. W. Johnson, "What are Emergent Properties and How Do They Affect the Engineering of Complex Systems?" *Reliability Engineering and System Safety*, Elsevier, Vol. 91, pp. 1475-81, 2006.

[Lap07] J. C. Laprie, K. Kanoun, M. Kaâniche, "Modelling Interdependencies between the Electricity and Information Infrastructures", *Lectures Notes in Computer Science, Computer Safety Reliability and Security*, Springer, Volume 4680/2007, pp. 54-67, 2007.

[Lev95] N. Leveson, *Safeware : System Safety and Computers*. Addison-Wesley Professional, 1995.

[Lev04] N. Leveson. "A new accident model for engineering safer systems", *Safety Science*, Vol. 42 No. 4, 237-270, Elsevier, 2004.

[Mas10] M. Masera, "Governance: How to Deal with ICT Security in the Power Infrastructure?" In *Securing Electricity Supply in the Cyber Age*, Springer, pp.111-127, 2010.

[Mad09] A. M. Madni, S. Jackson, "Towards a conceptual framework for resilience engineering," *IEEE Systems Journal*, vol. 3, no. 2, pp. 181-191, June 2009.

[NIST09] Office of the National Coordinator for Smart Grid Interoperability. *NIST Framework and Roadmap for Smart Grid Interoperability Standards* Release 1.0 (Draft). 2009.

[Ouy09] M. Ouyang, L. Hong, M. Zi-Jun, Y. Ming-Hui, Q. Fei, "A Methodological Approach to Analyze Vulnerability of Interdependent Infrastructures", *Simulation Modeling and Practice Theory*, Elsevier, Vol. 17, Issue 5, pp. 817-828, 2009.

[Pan08] S. Panzieri, R. Setola, "Failure Propagation in Critical Interdependent Infrastructures", International Journal in Modeling identification and Control, *Inderscience Publisher* Vol. 3, No. 1, pp.69-78, 2008.

[Per07] C. Perrow, *The Next Catastrophe: Reducing Our Vulnerabilities to Natural, Industrial, and Terrorist Disasters*. Princeton University Press, April 2007.

[Pru07] E. Pruyt, W. Thissen, "Transition of the european electricity system and system of systems concepts" in *IEEE Int. Conf. on Systems of Systems Engineering, SoSE'07*, September 2007.

[Rin01] S. M. Rinaldi, J. P. Peeremboom, and T. K. Kelly, "Identiyfing, Understanding and Analyzing Critical Infrastructure Interdependencies", *IEEE Control Systems Mag*., Vol. 21, no.6, pp. 11-25, 2001.

[Sha96] M. Shaw, D. Garlan, *Software Architecture: Perspectives on an Emerging Discipline*, Prentice-Hall, 1996.

[Sou08] A. Sousa-Poza., S. Kovacic, C. Keating, "System of Systems Engineering: an Emerging Multidiscipline", *International Journal of System of Systems Engineering*, Inderscience Publisher, Vol. 1, pp. 1 – 17, 2008.

[Sto96] N. Storey, *Safety Critical Computer Systems*. Addison Wesley, August 1996.

[Tyr05] J. Tyree, A. Akerman, "Architecture decisions: Demystifying architecture," *IEEE Software*, vol. 22, no. 2, pp. 19-27, 2005.

[UML10] *The Unified Modelling Language*. http://www.uml.org/ . Last visited on july 4, 2010.

[Val08] R. Valerdi, and others, "A Research Agenda for System of Systems Engineering", Int. J. System of Systems Engineering, *Inderscience Publisher*, Vol. 1, pp. 171-188, 2008.

European Commission

EUR 24727 EN – Joint Research Centre – Institute for the Protection and Security of the Citizen
Title: Infrastructure (Resilience-oriented) Modelling Language: I®ML
A proposal for modelling infrastructures and their connections
Authors: Andres Silva, Roberto Filippini
Luxembourg: Publications Office of the European Union
2011 – 32 pp. – 21x 29.7 cm
EUR – Scientific and Technical Research series – ISSN 1018-5593
ISBN 978-92-79-19324-8
doi:10.2788/54708

Abstract
The modelling of critical infrastructures (CIs) is an important issue that needs to be properly addressed, for several reasons. It is a basic support for making decisions about operation and risk reduction. It might help in understanding high-level states at the system-of-systems layer, which are not ready evident to the organisations that manage the lower level technical systems. Moreover, it is also indispensable for setting a common reference between operator and authorities, for agreeing on the incident scenarios that might affect those infrastructures. So far, critical infrastructures have been modelled ad-hoc, on the basis of knowledge and practice derived from less complex systems. As there is no theoretical framework, most of these efforts proceed without clear guides and goals and using informally defined schemas based mostly on boxes and arrows. Different CIs (electricity grid, telecommunications networks, emergency support, etc) have been modelled using particular schemas that were not directly translatable from one CI to another. If there is a desire to build a science of CIs it is because there are some observable commonalities that different CIs share. Up until now, however, those commonalities were not adequately compiled or categorized, so building models of CIs that are rooted on such commonalities was not possible. This report explores the issue of which elements underlie every CI and how those elements can be used to develop a modelling language that will enable CI modelling and, subsequently, analysis of CI interactions, with a special focus on resilience.

How to obtain EU publications

Our priced publications are available from EU Bookshop (http://bookshop.europa.eu), where you can place an order with the sales agent of your choice.

The Publications Office has a worldwide network of sales agents. You can obtain their contact details by sending a fax to (352) 29 29-42758.

The mission of the JRC is to provide customer-driven scientific and technical support for the conception, development, implementation and monitoring of EU policies. As a service of the European Commission, the JRC functions as a reference centre of science and technology for the Union. Close to the policy-making process, it serves the common interest of the Member States, while being independent of special interests, whether private or national.

LB-NA-24727-EN-C